Modern Engineering

Day 12

Prudential Financial

Whilst we wait....

In chat write:

Your name - role and a recent impulse buy.

Let's give everyone a min or two to join

Selenium Testing

LESSON ROADMAP

Browser Testing — Selenium — Testing our App with Selenium Lab

MEF MODULE 4 DAY 12: Selenium

Schedule	
9:00–9:15 am	Welcome and Warm-Up
9:15–9:45 am	End to End Testing and Browser Simulation
9:45 am–11:00 am	**Selenium Walkthrough**
11:00–12:30 pm	**Selenium Lab: Logging in to an Application**
12:30–1:30 pm	Lunch
1:30-1:30 pm	Testing with Selenium
1:30–2:30 pm	**Selenium Testing the To Do App**
2:30 - 4:50 pm	**Selenium Test your Express/React App**
4:50–5:00 pm	Bring It Home

LEARNING OBJECTIVES

1 Write automated browser tests using Selenium

2 Apply Selenium to a react application

3 Articulate strategies for balancing unit, integration and end to end tests

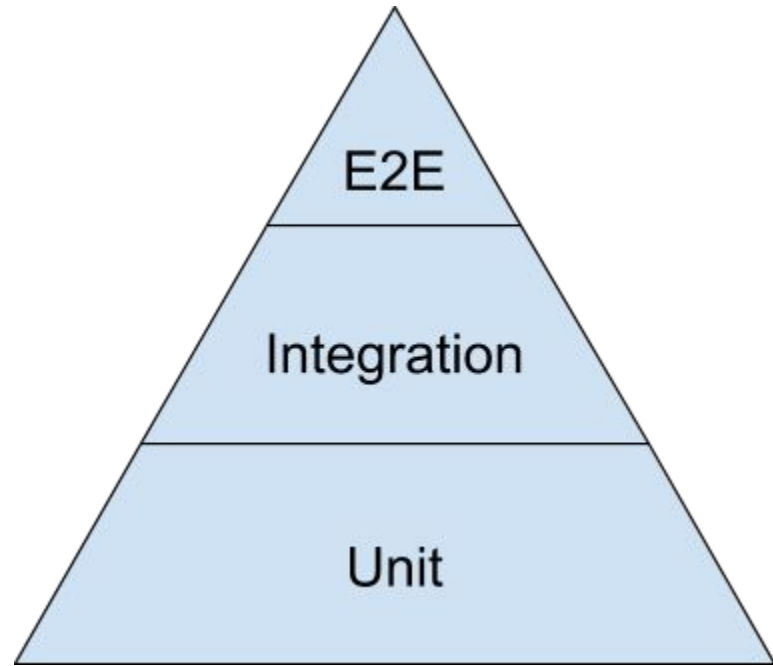**4** Combine multiple levels of testing into a cohesive testing suite

Modern Engineering

End to End Browser Testing

GA

E2E Testing: The Cherry on Top

Once all our unit and integration tests pass, it's finally time to add some End-to-End testing simulating the experience of a user.

Google recommends a 70-20-10 proportion of Unit, Integration, and End-to-End tests.

Source: Google Blog / Keynote Presentation

Selenium: Let the Robots Do the Clicking!

While we want to test the experience of users clicking and browsing through our app, this doesn't mean we have to do so manually.

The Selenium WebDriver library allows us to programmatically interact with the browser environment.

Let's head over to this git repository to start using Selenium.

Selenium Clicking Lab

Break Time

Modern Engineering

Testing with Selenium

Let's add some browser testing to the current test suite for our to-do application using Selenium

[Click Here To Join In!](#)

Now that we've seen how Selenium integrates with our previously written tests, it's time to add some browser UI tests to your own application.

One downside of writing browser tests is locking yourself into the UI you're writing tests for. If you want to change a button or layout, you end up having to also re-write the tests associated with it!

Therefore, you should prioritize the parts of your front-end you feel most confident in.